

Note

- Instructions have been included for each segment. You do not have to follow them exactly, but they are included to help you think through the steps.

```
In [15]: # Dependencies and Setup
import pandas as pd

# File to Load (Remember to Change These)
file_to_load = "Resources/purchase_data.csv"

# Read Purchasing File and store into Pandas data frame
purchase_data = pd.read_csv(file_to_load)
purchase_data.head()
```

Out[15]:

	Purchase ID	SN	Age	Gender	Item ID	Item Name	Price
0	0	Lisim78	20	Male	108	Extraction, Quickblade Of Trembling Hands	3.53
1	1	Lisovynya38	40	Male	143	Frenzied Scimitar	1.56
2	2	Ithergue48	24	Male	92	Final Critic	4.88
3	3	Chamassasya86	24	Male	100	Blindscythe	3.27
4	4	Iskosia90	23	Male	131	Fury	1.44

Player Count

- Display the total number of players

```
In [16]: total_players = len(purchase_data["SN"].unique())
Total_players = pd.DataFrame({"Total Players" : [total_players] })
Total_players
```

Out[16]:

Total Players
0
576

Purchasing Analysis (Total)

- Run basic calculations to obtain number of unique items, average price, etc.
- Create a summary data frame to hold the results
- Optional: give the displayed data cleaner formatting
- Display the summary data frame

```
In [17]: Unique_item = len(purchase_data["Item Name"].unique())
avarage = round(purchase_data['Price'].mean(),2)
number_of_purchase = len(purchase_data["Item Name"])
Total_revenue = purchase_data['Price'].sum()
summary_table = {"Number of Unique Item": [Unique_item],
                  "Average Price($)": [avarage],
                  "Number of Purchases" : [number_of_purchase],
                  "Total Revenue($)" : [Total_revenue]
                  }
Summary_table_Df = pd.DataFrame(summary_table)
Summary_table_Df
```

Out[17]:

	Number of Unique Item	Average Price(\$)	Number of Purchases	Total Revenue(\$)
0	179	3.05	780	2379.77

Gender Demographics

- Percentage and Count of Male Players
- Percentage and Count of Female Players
- Percentage and Count of Other / Non-Disclosed

```
In [23]: gender_count = purchase_data.groupby(['Gender'])['SN'].nunique()
gender_percentage = round(gender_count*100/total_players,2)
summary_table ={"Total Count": gender_count,
                 "Percentage of Players": gender_percentage,
                 }
Gender_Demographics = pd.DataFrame(summary_table)
Gender_Demographics
```

Out[23]:

	Total Count	Percentage of Players
Gender		
Female	81	14.06
Male	484	84.03
Other / Non-Disclosed	11	1.91

Purchasing Analysis (Gender)

- Run basic calculations to obtain purchase count, avg. purchase price, avg. purchase total per person etc. by gender
- Create a summary data frame to hold the results
- Optional: give the displayed data cleaner formatting
- Display the summary data frame

```
In [22]: purchase_count = purchase_data.groupby('Gender')['Purchase ID'].count()
Avarage_count = round(purchase_data.groupby('Gender')['Price'].mean(),2)
total_purchase = purchase_data.groupby('Gender')['Price'].sum()
total_avg_purchase = purchase_data.groupby(['SN', 'Gender'])['Price'].sum()
total_avg_per_person = round(total_avg_purchase.groupby('Gender').mean(),2)
purchase_analysis = pd.DataFrame({"Purchase Count": purchase_count ,
                                "Average Purchase Price ($)": Avarage_count,
                                "Total Purchase Value ($)": total_purchase,
                                "Avg Total Purchase per Person ($)": total_a
                                vg_per_person
                                })
purchase_analysis
```

Out[22]:

	Purchase Count	Average Purchase Price (\$)	Total Purchase Value (\$)	Avg Total Purchase per Person (\$)
Gender				
Female	113	3.20	361.94	4.47
Male	652	3.02	1967.64	4.07
Other / Non-Disclosed	15	3.35	50.19	4.56

Age Demographics

- Establish bins for ages
- Categorize the existing players using the age bins. Hint: use `pd.cut()`
- Calculate the numbers and percentages by age group
- Create a summary data frame to hold the results
- Optional: round the percentage column to two decimal points
- Display Age Demographics Table

```
In [24]: bins = [0,9,14,19,24,29,34,39,100]
Age_category = ["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40+"]
pd.cut(purchase_data['Age'], bins, labels=Age_category).head()
purchase_data["Age_group"] = pd.cut(purchase_data["Age"], bins, labels=Age_cat
egory)
group_data = purchase_data.groupby(['SN', 'Age_group'])['Age'].count()
total_counts = group_data.groupby(['Age_group']).count()
percentage = round(total_counts*100/total_players,2)
age_demographics = pd.DataFrame({"Total Count" : total_counts,
                                "Percentage of Players" : percentage})
age_demographics
```

Out[24]:

	Total Count	Percentage of Players
Age_group		
<10	17	2.95
10-14	22	3.82
15-19	107	18.58
20-24	258	44.79
25-29	77	13.37
30-34	52	9.03
35-39	31	5.38
40+	12	2.08

Purchasing Analysis (Age)

- Bin the purchase_data data frame by age
- Run basic calculations to obtain purchase count, avg. purchase price, avg. purchase total per person etc. in the table below
- Create a summary data frame to hold the results
- Optional: give the displayed data cleaner formatting
- Display the summary data frame

```
In [25]: bins = [0,9,14,19,24,29,34,39,100]
Age_category = ["<10", "10-14", "15-19", "20-24", "25-29", "30-34", "35-39", "40+"]
pd.cut(purchase_data['Age'], bins, labels=Age_category).head()
purchase_data["Age_group"] = pd.cut(purchase_data["Age"], bins, labels=Age_cat
egory)
purchase_count1 = purchase_data.groupby('Age_group')['Purchase ID'].count()
Avarage_count1 = round(purchase_data.groupby('Age_group')['Price'].mean(),2)
total_purchase1 = round(purchase_data.groupby('Age_group')['Price'].sum(),2)
total_avg_purchase = purchase_data.groupby(['SN', 'Age_group'])['Price'].sum()
total_avg_per_person1 = round(total_avg_purchase.groupby('Age_group').mean(),2
)
purchase_analysis_age = pd.DataFrame({"Purchase Count": purchase_count1 ,
                                     "Average Purchase Price($)": Avarage_count1,
                                     "Total Purchase Value($)": total_purchase1,
                                     "Avg Total Purchase per Person($)": total_av
g_per_person1
                                     })
purchase_analysis_age
```

Out[25]:

	Purchase Count	Average Purchase Price(\$)	Total Purchase Value(\$)	Avg Total Purchase per Person(\$)
Age_group				
<10	23	3.35	77.13	4.54
10-14	28	2.96	82.78	3.76
15-19	136	3.04	412.89	3.86
20-24	365	3.05	1114.06	4.32
25-29	101	2.90	293.00	3.81
30-34	73	2.93	214.00	4.12
35-39	41	3.60	147.67	4.76
40+	13	2.94	38.24	3.19

Top Spenders

- Run basic calculations to obtain the results in the table below
- Create a summary data frame to hold the results
- Sort the total purchase value column in descending order
- Optional: give the displayed data cleaner formatting
- Display a preview of the summary data frame

```
In [26]: purchase_count2 = purchase_data.groupby('SN')['Purchase ID'].count()
Avarage_count2 = round(purchase_data.groupby('SN')['Price'].mean(),2)
total_purchase2 = round(purchase_data.groupby('SN')['Price'].sum(),2)
total_spenders = pd.DataFrame({"Purchase Count": purchase_count2 ,
                                "Average Purchase Price($)": Avarage_count2,
                                "Total Purchase Value($)": total_purchase2
                                })

ordered = total_spenders.sort_values(by='Total Purchase Value($)', ascending=False)
ordered.head()
```

Out[26]:

	Purchase Count	Average Purchase Price(\$)	Total Purchase Value(\$)
SN			
Lisosia93	5	3.79	18.96
Idastidru52	4	3.86	15.45
Chamjask73	3	4.61	13.83
Iral74	4	3.40	13.62
Iskadarya95	3	4.37	13.10

Most Popular Items

- Retrieve the Item ID, Item Name, and Item Price columns
- Group by Item ID and Item Name. Perform calculations to obtain purchase count, item price, and total purchase value
- Create a summary data frame to hold the results
- Sort the purchase count column in descending order
- Optional: give the displayed data cleaner formatting
- Display a preview of the summary data frame

```
In [27]: new_purchase_data = purchase_data[['Item ID','Item Name','Price']]
grouped_data = new_purchase_data.groupby(['Item ID','Item Name'])
Item_price = grouped_data['Price'].first()
popular_items = grouped_data['Item ID'].count()
total_purchase_price = grouped_data['Price'].sum()
most_popular_items = pd.DataFrame({"Purchase Count": popular_items ,
                                   "Item Price($)": Item_price,
                                   "Total Purchase Value($)" : total_purchase_
price})
decending_ordered = most_popular_items.sort_values(by='Purchase Count', ascend
ing=False)
decending_ordered.head()
```

Out[27]:

		Purchase Count	Item Price(\$)	Total Purchase Value(\$)
Item ID	Item Name			
178	Oathbreaker, Last Hope of the Breaking Storm	12	4.23	50.76
145	Fiery Glass Crusader	9	4.58	41.22
108	Extraction, Quickblade Of Trembling Hands	9	3.53	31.77
82	Nirvana	9	4.90	44.10
19	Pursuit, Cudgel of Necromancy	8	1.02	8.16

Most Profitable Items

- Sort the above table by total purchase value in descending order
- Optional: give the displayed data cleaner formatting
- Display a preview of the data frame

```
In [28]: decending_ordered_purchase_value = most_popular_items.sort_values(by='Total Purchase Value($)', ascending=False)
decending_ordered_purchase_value.head()
```

Out[28]:

		Purchase Count	Item Price(\$)	Total Purchase Value(\$)
Item ID	Item Name			
178	Oathbreaker, Last Hope of the Breaking Storm	12	4.23	50.76
82	Nirvana	9	4.90	44.10
145	Fiery Glass Crusader	9	4.58	41.22
92	Final Critic	8	4.88	39.04
103	Singed Scalpel	8	4.35	34.80

In []:

In []: